Networking Foundations for Collaborative Computing at Internet Scope*

H.-Peter Dommel
Santa Clara University
Computer Engineering
500 El Camino Real
Santa Clara, CA 95053-0566
Email: hpdommel@scu.edu
Fax: (408) 554-5474

J. J. Garcia-Luna-Aceves
School of Engineering
University of California
Santa Cruz, CA 95064, USA
Email: jj@cse.ucsc.edu
Fax: (831) 459-4829

Abstract

Despite significant proliferation of Internet services in recent years, technology for computer-supported cooperative work and groupware have not progressed at the same rate. A wider distribution of the work force motivates the need for networked multimedia and groupware at Internet scope and for larger groups of end-users. In particular, synchronous telecollaboration enables people in different geographic locations to bridge time and space by sharing and jointly manipulating multimedia information in real-time and at various levels of granularity. This aspect stands in contrast to legacy client-server applications such as Internet radio broadcast or video-on-demand, and to asynchronous, document-centric collaboration tools like email, instant messaging, or chat rooms. In this paper, we provide a framework for network-supported synchronous multimedia groupwork at Internet scope and for large user groups. Contributions entail an novel classification for such systems concerning scale and scope of interaction, a formal framework for Internet sessions and mediation of access to concurrently shared resources, a taxonomy of crucial elements in cooperative applications, and a discussion of a generic network coordination protocol to sustain live interaction among concurrently active user groups. The core ideas put forward in this paper are useful for the characterization and rapid prototyping of a new generation of collaborative applications.

Keywords: Group coordination, Web-centric collaboration, Internet-wide Computer Supported Cooperative Work

1 INTRODUCTION

In contrast to stand-alone applications, where the user interacts only with a computer system, engineering of telecollaborative systems is much more complex because it involves user, network, and host-related issues, such as human factors, Quality-of-Service, and heterogeneous platforms for applications. The end-to-end interaction manifests itself between users, not end hosts, and users expect ideally a telecollaboration environment providing a quality of interaction close to a face-to-face meeting. Limitations in the availability and accessibility of resources in the shared workspace of a telecollaborative system create contention, competition, and conflict among users and make it necessary to deploy coordination mechanisms to reach consensus on how to jointly and effectively use the resources. Conflicts stalling the workflow may occur before and during resource allocation to users, as well as during actual usage. Telecollaborative services build on the provision of group coordination mechanisms. These manage access, manipulation, distribution and presentation issues between users and shared resources. Such coordination mechanisms are necessary to allow users to achieve individual goals in the context of group-centered remote interaction, when *telepresence* [3] substitutes for physical presence. Cerf *et al.* [6] pointed out the importance of transatlantic collaboration infrastructures in a memorandum in 1991.

Software to support collaborative work, generally termed *groupware* [11], or workgroup computing software, referred initially only to systems supporting the asynchronous exchange of text-documents, but more recent connotations include multimedia-based, synchronous interaction. We focus on group coordination protocols, which embrace multicasting and consider network conditions in the coordination processes between hosts, complementing efforts on group membership known from distributed systems and multicasting as an efficient message dissemination mechanism for group communication. This extended abstract presents an outline of our work on a group coordination architecture, with focus on the formal modeling of the key elements and protocols. Section 2 discusses the group coordination framework and main architectural aspects and outlines a generic coordination protocol. Section 3 concludes the paper.

2 COORDINATION FRAMEWORK

We present a formal view on entities and actions in coordination-centric systems, refining earlier efforts [24, 25] on the definition of coordination and control processes

*This work was supported in part by the Defense Advanced Research Projects Agency (DARPA) under Grants F19628-96-C-0038 and DAAB07-97-C-D607.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **2006**	2. REPORT TYPE	3. DATES COVERED **00-00-2006 to 00-00-2006**
4. TITLE AND SUBTITLE **Networking Foundations for Collaborative Computing at Internet Scope**		5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of California at Santa Cruz,Department of Computer Engineering,Santa Cruz,CA,95064**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**7**	

in collaborative multimedia systems. Candan *et al.* [5] focus on algorithms for collaborative composition and transmission of media objects under given quality constraints, and their presentation in collaborative group-sessions. We picture a computer network as a graph with nodes (stations, hosts) V sending messages across links (channels) $E \subset V \times V$. A connection is a unidirectional or bidirectional transmission link from a sender node to a set of receiver nodes.

Definition 1 *A collaboration environment* Γ *in a computer network is a tuple*

$$\Gamma =< \mathcal{S}, \mathcal{U}, \mathcal{R}, \mathcal{F} >$$

where $\mathcal{S} = (V, E)$ *is a set of sessions* Σ*,* $\mathcal{U}$ *is a set of users (hosts, processes, agents, participants),* $\mathcal{R}$ *is a set of shared resources (media), and* $\mathcal{F}$ *is a set of floors controlling the resources.*

2.1 Entities

2.1.1 Sessions

A session provides the infrastructure for cooperation and collaboration.

Definition 2 *A session* $\Sigma \in \mathcal{S}$ *is a tuple*

$$\Sigma =< Sid, T_i, T_e, A_S, L >$$

where Sid *is a unique identifier within* Γ*,* T_i *is the initiation or announcement time,* T_e *is the ending time, and* A_S *is a list of attributes characterizing the session at level* L*. A conference is a set of sessions* $\Sigma_i \in \mathcal{S}$*, where* $i \geq 1$*.*

Sid is a unique session identifier per collaborative environment, whose sequence number space is wrapped around in correlation with the turnover rate and lifetime of sessions in Γ. The time may reflect real-time, logical time, or define a lifetime interval $\Delta = T_e - T_i$. L denotes the session level (default 0).

$A_S = (M, O, C)$ describes purpose and orchestration of a session in terms of membership M, organization O, and control C, as shown in Figure 1. Szyperski [31] characterizes session types in a similar, but less refined way, according to the model of interaction (controlled, dynamic, static) and data flow (1-n, n-1, m-n). For instance, a lecture is a controlled, long-term interaction between one sender and n receivers. A typical $n-1$ session is telemetry, and a whiteboard session is typically $m-n$. Our session characterization applies to specific collaborative applications, as well as generic session types in the spectrum of real-time collaborative work, such as lectures, business meetings, labs, panels, brainstorm meetings, exams, interviews, or chats.

Membership reflects the composure of the user group in the session. *Participation* specifies whether information is exchanged unilaterally, or bilaterally relative to a host, impacting user access rights and data-flow. Interactive sessions may be symmetric, i.e., all users have the same view on shared resources (WYSIWIS), or asymmetric, where users pertain individual views on the same shared data space (relaxed WYSIWIS) [29]. *Size* specifies a small (< 5), medium (< 100), or large (≥ 100) number of users, impacting scalability of the coordination mechanism. *Accessibility* declares whether a session is open, allowing any

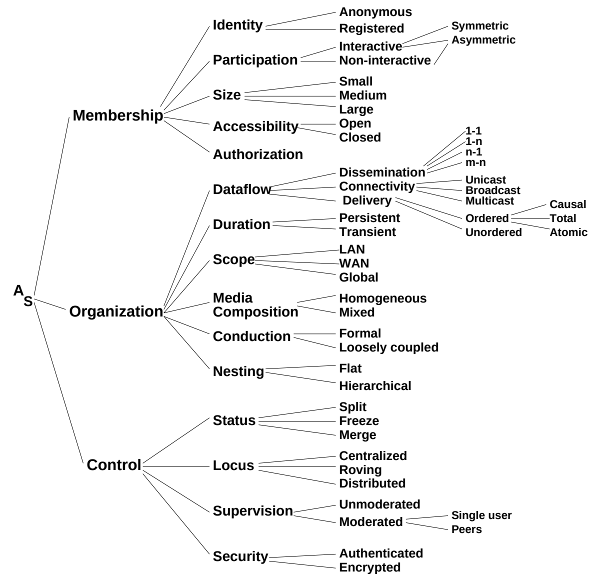

Figure 1. Session attributes.

user to join, whereas closed sessions allow participation by invitation only. Authorization specifies whether coordination primitives may use read-only, read-write, or write-only privileges for the entire session. Users may have individual, role-based authorizations, as well.

Organization entails specifics on how the session is to be orchestrated. *Dataflow* describes how data are multiplexed among users, with a 1-1, 1-n, or 1-m transmission model and with unicast, broadcast, or multicast in a session of n users, where $m \leq n$. Delivery can be ordered or unordered. it Duration discerns between sessions with longer lifetime (persistent) vs. short-term sessions, where the precise timing modalities are case-specific and left open. *Scope* specifies the hop limit for packets sent by hosts in a particular session, similar to the Time-To-Live semantics in IP, which allows to constrain sessions to a geographic range and retain privacy or limited dissemination to a specific group. *Media composition* defines whether the session uses a single medium such as audio-only, or mixed media, e.g., a video-audio combination. *Conduction* refers to the session agenda and moderation style, which can be either tightly coupled, i.e., all users know about each other and follow some agenda in the style of “Robert’s Rules of Order” [26], or the exchange is loosely-coupled and not prescribed. Sessions can be flat ($L = 1$) or maintain two or more levels with *nested* groups ($L \geq 1$).

Control depicts the status, locus of control, and security measures activated for a session. Sessions with overlapping or diverging interests can merge or split. Such reconfiguration of sessions with regard to membership and session events linked to specific phases must be possible without session termination or restart of applications. The session *status* marks, whether the session is a partition from a larger session, frozen but still deemed as active, merged or revived. Tracking of states in coordination protocols and the outcome of coordination processes can be logged and persistent, or ephemeral.

Locus of control specifies, whether membership and floor control are being handled in one central location, partially distributed among several servers, or fully distributed across all hosts. Partial or full replication is possible for the latter two paradigms. A central controller can also rove among all sites and achieve better fault tolerance. Distributed control is multilateral, with varying degrees of “consentience” and “equipollence”, i.e., how much everybody participates and how authorities and responsibilities are allocated. Multilateral control is either successive, partitioned, democratic or anarchic. Successive controllership allows one distinct controller at a time, and alternates among users, and partitioned control lets several controllers each perform a subset of control operations. Democratic control lets all users contribute to the control process, e.g., via voting. Anarchic control gives all subjects complete freedom of acting and control of sharing is performed peer-to-peer based.

The locus of control is related to the *supervision* attribute, indicating whether the communication process in coordination is moderated, peer-reviewed, or free. A moderator decides which users may send information, what is forwarded to the receivers, or which receivers may receive a particular content or access a specific resource, implementing a notion of floor control. McKinlay *et al.* [20] note for face-to-face meetings that the importance of chaired guidance increases with the session size, and the difficulty in performing a joint task, since each member's ability to participate and influence others is reduced. Finally, coordination touches upon *security* issues, specifying whether users are anonymous or authenticated in their exchanges, either at session initiation, or at every turn, and whether information is encrypted.

2.1.2 Hierarchical Sessions

Rajan *et al.* [24] identify a *confluence* as a special session type, where all participants transmit and receive the same set of media streams mixed together in broadcast, which allows to save bandwidth. The notion of confluences and session nesting leads to the concept of multilevel or *hierarchical* sessions. Session hierarchies permit aggregation of users at various levels of abstraction, reflecting interests, the stage of task completion, authorizations, temporary subgroups (coteries), or geographic proximity, and reflects the inherently hierarchical group dynamics of face-to-face meetings better. The hierarchy is denoted with the session level parameter L, which indicates numerically the position of Sid in a session hierarchy. For instance, in a 3-level hierarchy, a collaboration or master session has level 0, a session level has level 1, and a subsession is at level 2, which may be sufficient to characterize most collaboration scenarios.

Rangan and Vin [25, 34] give formal definitions for collaborative systems including conference, session and stream abstractions for the purpose of automated reasoning about the properties of multimedia collaborations. Adopting their definitions to the session context, we distinguish between *simple sessions* containing individual users, and *super sessions*, recursively consisting of other sessions $s_{i,L} \in S_i$ and individuals, with L indicating the level of membership. We denote the outmost “root” session as level-0 session. Many conference scenarios contain only two sublevels, *subsessions* with $L = 1$ and *coteries* with $L = 2$. Coteries permit private subgrouping for brief exchanges (“sidechats”) [36] without requiring its members to leave the larger group context or open a separate multicast group. Neilsen and Mizuno describe a membership algorithm for joining and leaving coteries [21], and Texier and Plouzeau [32] propose object binding algorithms for multiple sessions, however, to date a sound mechanism for session management in multimedia collaboration is still missing. *Concurrent* sessions, as opposed to sequential sessions, allow users to participate in multiple sessions simultaneously. *Hierarchical sessions* permit inheritance of attributes from parent to child sessions, and aggregation of sibling sessions under a parent session. Vin *et al.* [35] describe such a hierarchical architecture for media mixing, as required in a telepresentation system of teleorchestra, and derives upper bounds for the media transmission capacity and the height of a hierarchy, given a number of participants and mixers, with one speaker being active at a time.

2.1.3 Users

Users in the user set $\mathcal{U}$ from the specification of a collaborative environment, also referred to as participants, subjects, or session members, are equated with hosts or their processes in protocol descriptions.

Definition 3 *A user $U \in \mathcal{U}$ is a tuple*

$$U =< Uid, Sid, Loc, T_j, T_l, A_U >$$

where Uid is a unique identifier within the session Sid, Loc is the local or remote location, given as IP-address or unique host identifier, T_j is the joining time, T_l is the leaving time, and A_U is a list of user attributes.

Processes can be system agents [10, 19] executing on behalf of a user. The user attributes A_U are depicted in Figure 2.

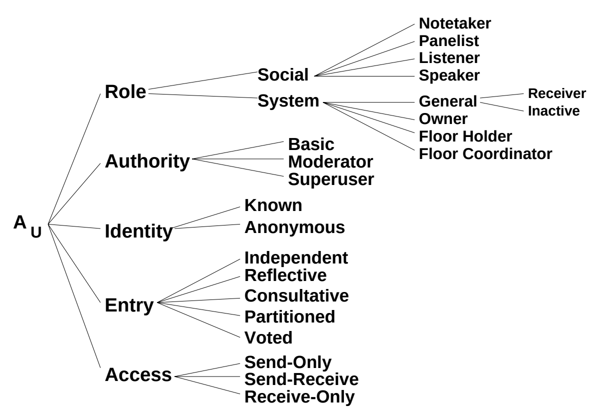

Figure 2. User attributes.

Accordingly, users are characterized by their roles, authority, identity, entry capabilities and access rights, which decide the floor control strategy applicable. Users can be co-located in the same space, or geographically distributed. We distinguish between social and system **roles**. *Social roles* describe the function of a user within a session, e.g., being a panelist or lecturer. *System roles* refer to the control function within a floor control protocol: participants without specific role can be receiver or inactive; the owner of a resource $r \in R$ is the node that injects r into a session and initiates floor control for r, which may vanish from a session if the owner leaves; the floor coordinator (FC) is an arbiter over a resource r, or a session moderator granting or denying a floor on r during session time to the floor holder (FH), who attains the exclusive right to work on r for a floor holding period. FC and FH may be located at different nodes, or be assumed by the same node. These roles may be statically assigned at session start, or rove among users during session conduction. Users without control roles are general session members, and can be active or inactive, depending on whether they invoke state transitions in the coordination mechanism.

A moderator is a special FC case, where the coordinator role is assigned to a user to supervise content exchange, resource usage, and membership for a specific section of the full lifetime of a session. A moderator-driven session, mediated through a specific host, results in a centralized coordination scheme, even though the host topology may be decentralized, with the known shortcomings in regard to efficiency and resiliency. Moderators may be selected, because they start a session or are chosen by session members in advance, or they may be elected [13, 28] at session runtime. Tijdeman [33] discusses a solution for the chairman assignment problem such that at any time the accumulated number of chairmen from each state (or session) is proportional to its relative weight. Role-based floor control in dynamic sessions contrasts static *role-based access control* (RBAC) models [27]. Roles can be inherited from a supersession to a subsession.

Authority defines, whether the user is a simple participant, privileged as system root user, or moderator, linking this field with the role entries. A moderator can be permanent FC. As social role, the moderator equates to a session supervisor being able to inspect all session turns between users. *Identity* specifies whether the user wants to remain anonymous or whether the Uid can be posted to the session. An *Entry* is either independent, i.e., unaware of the actions and entries of others, reflective, i.e., polling session members, consultative based on "contextual clue messages", partitioned and representing a subtask, based on voting among the group, or debriefed and recorded [11]. In addition, user entries may be temporary or permanent, and logged for the purpose of reviewing histories of collaborative sessions, or undoing certain steps [23]. *Access* defines the basic privileges to work on a resource, in receive-only, send-and-receive, and send-only mode, in analogy to read and write authorizations in file systems. We introduce the notion of a group to describe associations of users within sessions.

Definition 4 *A user group (multicast group) G is a set of users U with common session and user attributes, expressing a common media or task focus, such that $\mathcal{U} \supseteq G \supseteq U$.*

2.1.4 Resources

Multimedia collaborative systems use a polymorphic or multimodal mix of resources being shared across networks. A resource can be an application, host object, or network entity shared in collaboration at various levels of granularity. Four primary classes of multimedia traffic with different Quality-of-Service characteristics exist [1]: *control packets* for coordination information are mostly of low volume, but need reliable transmission; *real-time media* transport time-critical information and tolerate some loss; *elastic media* are apt for discrete information with relaxed timing constraints, but tolerate no loss; and *bulky media*, which require high throughput and reliable transmission, but can tolerate some delay. We define resources as application components in our coordination framework:

Definition 5 *A resource $R \in \mathcal{R}$ is a tuple*

$$R = < Rid, Sid, Pid, Uid, T_c, T_d, A_R >$$

where Rid is a unique resource identifier owned by user Uid within session Sid. Pid is the parent identifier or the resource that Rid belongs to, T_c is the time of creation or injection of the resource into the collaborative workspace, T_d is the deletion time, and A_R is a list of resource attributes.

Rid designates both discrete media and streaming media and may contain the port where the resource is transmitted. The resource attributes A_R are depicted in Figure 3. The Pid value allows for recursive subsumption of resource components within resources, and hence sharing or resource components at an arbitrary granularity. For instance, users can share an entire window, or a graphical object within that window.

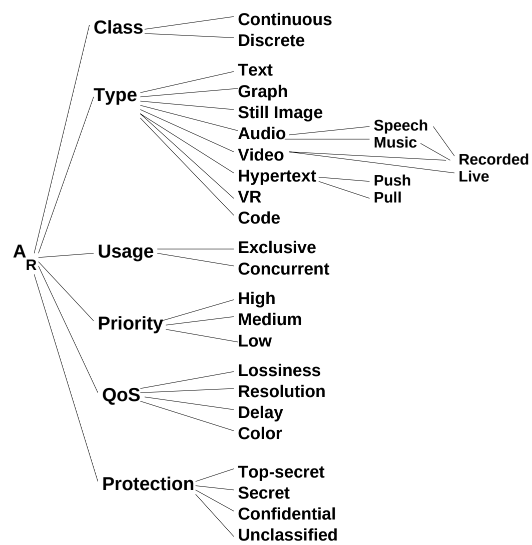

Figure 3. Resource attributes.

Class describes whether the resource is continuous or discrete. **Type** characterizes the media object class, indicating whether a resource is text-based, graphical, or some real-time medium and identifies the purpose it serves. Resources can be virtual, or they can represent actual remote devices, for example, a surgical instrument in telemedicine. Resources can be mixed and need not necessarily be proprietary to the session from which they are accessed, but could also be hosted on a machine "outside" of the session. Coordination on *text*, as the default medium for most collaborative system, revolves around alternate typing, for instance in chat tools, or concurrent editing from chapters or sections to single sentences. Text can be plain ASCII, or one or various rich text formats with formatting commands. *Graphics* tools, such as drawing and design tools, necessitate coordination in time and space, either by marking areas on a shared canvas or objects for shared editing, or by introducing graphical widgets such as telepointers. Functions the compute or render the shared workspace in a specific way, are another coordination component. *Still images* require also spatiotemporal coordination and allow for multiple image formats, such as TIFF, GIF, or JPEG. *Audio* tools, for speech, or music data such as MIDI require temporal coordination in recording and replay, and spatiotemporal coordination in editing. For instance, a shared audio channel or music stream requires sequenced access, whereas joint editing of a music score is a spatial aspect. Silence detection is useful for more efficient processing of audio streams, but also help to trigger speaker floor switching. *Video* concerns motion image display and editing, either from a live source, stored locally, or replayed on demand, and is often used in combination with audio, requiring temporal coordination. Various formats, such as H.263 or MPEG, should be supported. *Hypertext* information is multimodal and integrates all of the above resource types using for instance HTML or XML, and is either geared for server-push or client-pull. *VR* (Virtual Reality) [8, 7, 15] is similarly multimodal, but adds input and output devices giving the user three-dimensional orientation or tactile sensations. Coordination must be interfaced with collision control [17] in virtual spaces. *Code* comprises application-specific structured documents such as Postscript, MIME email [4], or LaTeX. A *device* is a hardware unit serving as access point, such as a camera. A *multimedia conference* is a conference using multimodal resource types.

Usage determines, if the resource can be used concurrently by multiple users or requires sequential processing with exclusive floors. For instance, a shared whiteboard allows for multiple concurrent telepointers with a small number of users, whereas a remotely controlled camera can only perform a positioning command for one user at a time. **Priority** sets an importance value on the transmission and processing of the information, preempting other media dissemination of lower ratings. **QoS** defines the required Quality-of-Service [30] for the resource, including the tolerable loss, the required resolution, the possible maximum delay, and the color depth. Other criteria may be added depending on the nature of the resource, such as the channel number, a frame-rate, encoding scheme, sampling rate etc. The **Protection** attributes denotes whether a resource is public, private, or proctored, which may be expressed with a numerical value, or work in analogy with the *Bell-LaPadula* model [2], discerning between top-secret, secret, confidential, or unclassified information [12]. The degree of security determines the required encryption level and method to prevent forgery of control states and coordination messages. In contrast to traditional models of protection giving access to a resource based on user identity, coordination-based access must take into account the task to be performed. Predominant measures to shield off internetworks with firewalls make real-time collaboration very difficult and are a major impediment in the realization of Internet collaboration. While new concepts for secure collaboration architectures are emerging [14], efficient key management and encryption in conjunction with floor control have yet to be developed.

Resources $r \in R$ can be located at one particular node, be distributed in their components across the node set, or be replicated over all nodes. Figure 4 depicts access paradigms for shared resources. In case (a), one or more resources are centralized and accessed by multiple parties; case (b)

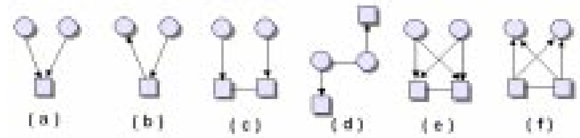

Figure 4. Resource access scenarios: (a) Centralization, (b) Producer-Consumer, (c) Replication, (d) Distribution, (e) Multi-resource access, (f) Multi-resource consumption.

lets one host produce a resource and other hosts consume it; case (c) shows the case, where each party maintains a replica of the same resource locally, exchanging updates on a regular basis; in case (d) all hosts maintain partial information on the shared resource, using a distributed protocol to aggregate the information; and cases (e) and (f) show access or consumption of multiple resources by multiple parties. These constellations are the baseline for configuring a coordination mechanism to adapt to various constellations of the shared workspace. A location mechanism for resources within sessions, and mapping scheme from resource objects to multicast groups is needed, as partially implemented with the CCCP protocol [16].

2.1.5 Floors

A floor is a temporary access and manipulation privilege for multimedia resources in interactive groupwork, generalized to the domain of CSCW from the "right to speak" [26]. A floor control protocol mediates access to shared objects by granting floors according to a group-specific service policy.

Definition 6 *A floor $F \in \mathcal{F}$ is a tuple*

$$F = < Fid, Rid, Uid, T_i, T_d, A_F >$$

where Fid is a unique floor identifier within the shared workspace for a resource Rid, assigned to user Uid at inception time T_i, and deactivated at time T_d, with A_F denoting a list of floor attributes.

Note that one Rid may have multiple Fid assigned for control of various granules, but each floor is controlling exactly one resource. Floors are indirectly associated with sessions via Rid, and floor properties may be inherited from a master resource to its subcomponents. We assume that one floor F is assigned per resource component. The pairing (Fid, Rid) specifies the granularity of control and the commands available with possession of the floor. A floor can control an entire conference, an application, a single window, or a shared object [18]. For instance, for audio the associated commands may be `talk, mute, pause`. Video floor commands are for instance `caption, forward, cut, replay`. Floors can be static relative to a session lifetime, or dynamic, i.e., assigned ad hoc by a computer or social protocol. The combination of Uid and the attributes specifies whether the user is FC, FH, chair, or general participant. T_i and T_d may ne set using real-time clocks, or a logical session time. Figure 5 depicts the floor attributes.

With regard to **directionality**, we discern between *sender floors* and *receiver floors*. A receiver floor refers

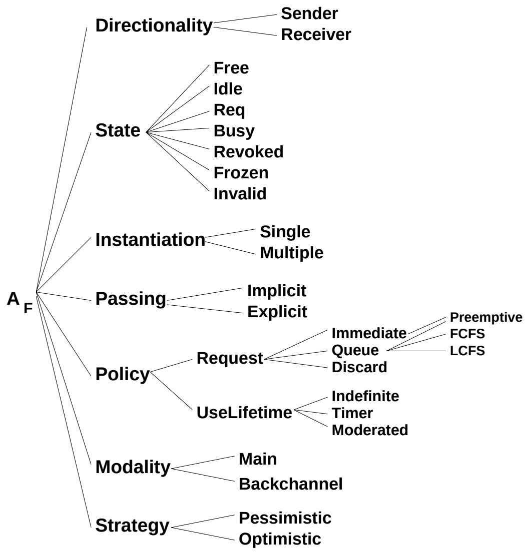

Figure 5. Floor attributes.

to the passive control concept that a user can filter or deny specific received streams ("What I See Is What I Want"). Floor control typically refers to source-based control, which may reduce traffic significantly ("What You See Is What I Share"). **State** defines the generic operational states of a floor control mechanism. *Free* denotes an available, unused floor, *Idle* denotes an assigned, but inactive floor, *Req* marks a floor as being requested, *Busy* is the tag for a granted and assigned floor, *Revoked* marks a floor, whose lifetime is shortened by a moderator or a preemption mechanism, *Frozen* marks a floor in a pending session, and *Invalid* identifies a nonexistent floor. A generic floor control protocol defining the transitions between these states is depicted in Figure 6.

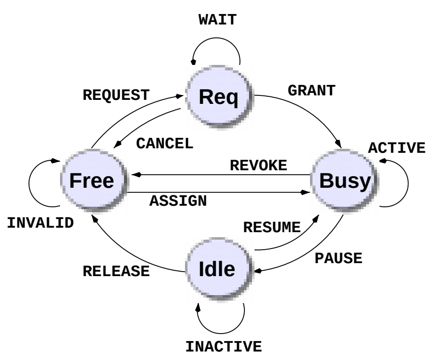

Figure 6. Generic floor control protocol.

Floor control can be relaxed for concurrent activities where the chance of direct conflict is smaller, e.g., in joint editing of text paragraphs, but it must be strict in opposing activities such as speaking over the same audio channel. **Instantiation** defines accordingly, how many instances of the same floor may exist concurrently in the system. A remote instrument with exactly one function to be shared permits a *single* floor, whereas telepointers on a whiteboard canvas may coexist in *multiple* renditions. Disjoint parties may receive multiple instances of a floor, e.g., user groups $(U_1, U_2)_A$ and $(U_3, U_4, U_5)_A$ may independently converse with an audio floor $F = A$.

Passing describes whether floor management is tangible or transparent to end-users. *Explicit* control gives handles to users to start and initiate turns based on the exchange of markers that signify possession of the floor, contrasting *implicit* control, where no beacons are exchanged to transfer floors. Control may follow a programmed session agenda, or allow for open interaction. Explicit control is manifested for instance by pressing the `Request` button in a shared application. Implicit control is realized by users observing inactivity on the resource and taking action when appropriate. **Policy** defines the request and usage rules. The *request* policy determines, whether floor requests are immediately satisfied, queued and served according a queuing policy, or discarded, when there is not opening for the floor. A chairperson may preempt any floor activity. *UseLifetime* denotes, whether a floor can be used indefinitely until being requested, or whether a timer or moderator control the duration of usage. **Modality** distinguishes between *main* floors assigned for primary communication from a sender to a receiver, of *backchannel* floors used to give brief feedback.

We can distinguish between four paradigms to deal with race conditions in cooperative work: *blocking* of conflicts with exclusive locks, *disallowing* of conflicts with permission tokens, *mitigating* conflicts by detecting dependencies and reordering of activities into non-conflicting series, and *resolving* of inconsistencies created through conflict. The first two paradigms are restrictive and prevent conflicts, the latter two are permissive and allow for progress into conflict with preconditions and postconditions. Therefore, the *strategy* entails *pessimistic* control following the premise of conflict avoidance, versus *optimistic* control as the strategy to allow conflicts and provide means such as dependency detection[29] to resolve them.

Previously [9] we proposed the idea to integrate group coordination services with the IP-multicast infrastructure and framework, which is currently gradually deployed on the Internet, so that coordination services should be deployed on top of reliable multicast and ideally operate on the same logical network topology. This approach eliminates the need for a separate control infrastructure for tracking, routing, withholding, or forwarding coordination directives and enables distributed activities in large groups and at large distances with low latencies.

The presented model serves both theoretical and practical purposes. It provides a more elaborate framework for formal specification and validation of collaborative systems, e.g., with the prototype verification system [24]. It also allows for session capability descriptions [22] for on-the-fly specification, set up and query of the membership and coordination status of an active conference. A capability is understood as a resources or system feature influencing the selection of useful configurations for components.

3 SUMMARY

Our objective is to explore the key elements for a new breed of coordination protocols and architectures useful for engineering computer-supported cooperative work applications operating at Internet scope. We view coordination as the third integral component in group-oriented communication services in the Internet, complementing group dissemination and membership protocols and enriching the current IP-multicast service model, which lacks refined support for group coordination. The goal of our framework is to characterize the relevant parameters in designing an API for rapid development of group coordination-supportive applications. To this end, we elaborated on a novel taxonomy of typical coordination components in collaborative multimedia applications.

References

[1] L. Aguilar, J. J. Garcia-Luna-Aceves, D. Moran, E. J. Craighill, and R. Brungardt. Architecture for a multimedia teleconferencing system. In *Proc. ACM SIGCOMM*, pages 126–136, Aug. 1986.

[2] D. E. Bell and L. J. L. Padula. Secure computer system: Unified exposition and multics interpretation. Technical Report MTR-2997, Mitre Corp., Bedford, MA,, July 1975.

[3] G. Bell. Telepresence. Presentation at UC/Berkeley, Feb. 1996.

[4] N. Borenstein and N. Freed. MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for specifying and describing the format of Internet message bodies. RFC 1521, Sept. 1993.

[5] K. S. Candan, V. S. Subrahmanian, and P. V. Rangan. Towards a theory of collaborative multimedia. In *Proc. 3rd IEEE Int. Conference on Multimedia Computing and Systems*, pages 279–282, Hiroshima, Japan, June 1996.

[6] V. Cerf, P. T. Kirstein, and B. Randell. Network and infrastructure user requirements for transatlantic research collaboration. RFC 1210, March 1991.

[7] C. Charlsson and O. Hagsand. DIVE - a platform for multi-user virtual environments. *Computers & Graphics*, 17(6):663–669, Nov.-Dec. 1993.

[8] E. F. Churchill and D. Snowdon. Collaborative virtual environments: an introductory review of issues and systems. *Virtual Reality*, 3(1):3–15, 1998.

[9] H.-P. Dommel and J. J. Garcia-Luna-Aceves. Group coordination support for synchronous Internet collaboration. *IEEE Internet Computing Magazine, Special Issue on Multimedia and Collaborative Computing over the Internet*, pages 74–80, Mar./Apr. 1999.

[10] E. A. Edmonds, L. Candy, R. Jones, and B. Soufi. Support for collaborative design: Agents and emergence. *Comm. of the ACM*, 37(7):41–47, July 1994.

[11] C. A. Ellis, S. J. Gibbs, and G. L. Rein. Groupware - some issues and experiences. *Comm. of the ACM*, 34(1):38–58, Jan. 1991.

[12] R. Elmasri and S. B. Navathe. *Fundamentals of Database Systems, 2nd Ed.* The Benjamin/Cummings Publ. Company, Redwood City et. al., 1994.

[13] H. Garcia-Molina. Elections in a distributed computing system. *IEEE Trans. On Computers*, 1(Jan.):48–59, C-31 1982.

[14] L. Gong. Enclaves: Enabling secure collaboration over the Internet. *IEEE Journal on Selected Areas in Communications*, 15(3):567–575, April 1997.

[15] C. Greenhalgh and S. Benford. MASSIVE: A collaborative virtual environment for teleconferencing. *ACM Trans. on Computer-Human Interaction*, 2(3):239–61, Sept. 1995.

[16] M. Handley, I. Wakeman, and J. Crowcroft. The Conference Control Channel Protocol (CCCP): A scalable base for building conference control applications. In *Proc. ACM SIGCOMM*, pages 275–287, Cambridge, MA, Aug./Sep. 1995.

[17] P. M. Hubbard. Collision detection for interactive graphics applications. *IEEE Trans. on Visualization and Computer Graphics*, 1(3):218–30, Sept. 1995.

[18] J. C. Lauwers and K. A. Lantz. Collaboration awareness in support of collaboration transparency: Requirements for the next generation of shared window systems. In *Proc. SIGCHI*, pages 303–311, Seattle, WA, Apr. 1990.

[19] K.-C. Lee, W. H. M. Jr., and A. P. Sheth. A framework of controlling cooperative agents. *IEEE Computer*, pages 8–16, July 1993.

[20] A. McKinlay, R. Procter, O. Masting, R. Woodburn, and J. Arnott. Studies of turn-taking in computer-mediated communications. *Interacting with Computers*, 6(2):151–171, June 1994.

[21] M. L. Neilsen and M. Mizuno. Coterie join algorithm. *IEEE Trans. on Parallel and Distributed Systems*, 3(5):582–590, Sept. 1992.

[22] J. Ott, D. Kutscher, and C. Bormann. Capability description for group cooperation. Internet Draft draft-ott-mmusic-cap-00.txt, June 1999.

[23] A. Prakash and M. J. Knister. A framework for undoing actions in collaborative systems. *ACM Trans. Computer-Human Interaction*, 1(4):295–330, Dec. 1994.

[24] S. Rajan, P. V. Rangan, and H. M. Vin. A formal basis for structured multimedia collaboration. In *Proc. 2nd IEEE Conf. on Multimedia Computing and Systems*, Wash., D.C., May 1995.

[25] P. V. Rangan and H. M. Vin. Multimedia collaboration as a universal paradigm for collaboration. In *Multimedia - Principles, Systems and Applications*, pages 3–15. Springer-Verlag, Apr. 1991.

[26] H. M. Robert. *Robert's rules of order*. Bantam Books, Toronto; New York, 1986.

[27] R. S. Sandhu and E. J. Coyne. Role-based access control models. *Computer*, 29(2):38–47, Feb. 1996.

[28] S. Singh and J. F. Kurose. Electing "good" leaders. *J. of Parallel and Distributed Computing*, 21(2):184–201, May 1994.

[29] M. Stefik, G. Foster, D. G. Brobrow, K. Kahn, S. Lanning, and L. Suchman. Beyond the chalkboard: Computer support for collaboration and problem solving in meetings. *Comm. ACM*, 30(1):32–47, Jan. 1987.

[30] R. Steinmetz and K. Nahrstedt. Multimedia*: Computing, Communication, and Applications*. Prentice Hall, Upper Saddle River, NJ, 1995.

[31] C. Szyperski and G. Ventre. A characterization of multiparty interactive multimedia applications. Technical Report TR-93-006, International Computer Science Institute, Berkeley, Feb. 1993.

[32] G. Texier and N. Plouzeau. Automatic management of sessions in shared spaces. In *Proc. Int. Conf. on Parallel and Distributed Processing Techniques and Applications (PDPTA'99)*, pages 67–73, Las Vegas, NV, June/July 1999.

[33] R. Tijdeman. The chairman assignment problem. *Discrete Mathematics*, 32(3):323–330, Dec. 1980.

[34] H. M. Vin and P. V. Rangan. System support for computer mediated multimedia collaboration. In *Proc. ACM CSCW*, pages 203–209, Nov. 1992.

[35] H. M. Vin, P. V. Rangan, and S. Ramanathan. Hierarchical conferencing architectures for inter-group multimedia collaboration. In *ACM SIGOIS Bull., Proc. Org. Comp. Sys.*, pages 43–54, Atlanta, GA, Nov 1991.

[36] H. M. Vin, P. T. Zellweger, D. C. Swinehart, and P. V. Rangan. Multimedia conferencing in the Etherphone environment. *Computer*, 24(10):69–79, Oct. 1991.